WEST Search History

DATE: Tuesday, July 29, 2003

Set Name side by side	Query	Hit Count	Set Name result set
DB=JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ			
L11	L9 and string	4	L11
L10	L9 and heap and string	0	L10
L9	object oriented operating	95	L9
DB=USPT; PLUR=YES; OP=ADJ			
L8	L7 and buffer	13	L8
L7	L6 and string and heap	25	L7
L6	object oriented operating	369	L6
L5	L4	14	L5
DB=USPT,PGPB; PLUR=YES; OP=ADJ			
L4	L3 and text string	17	L4
L3	L2 and pointer	226	L3
L2	L1 and buffer and heap	253	L2
L1	memory management	8736	L1

END OF SEARCH HISTORY

CiteSeer Find: buffer descriptor Documents Citations

Searching for **PHRASE object handle string**.
Restrict to: Header Title Order by: Citations Hubs Usage Date Try: Amazon B&N Google (RI) Google (Web) CSB DBLP
No documents match Boolean query. Trying non-Boolean relevance query.
1000 documents found. Retrieving documents... **Order: relevance to query.**

Scheduling Concurrent RPCs in the Globe Location Service - Ballintijn, Sandberg, van.. (1997) (Correct)
Keywords: concurrent RPC, distributed **objects**, wide-area computing Abstract Globe is a
an **object**, a process first retrieves its **object handle** from a name server or by other means. An **object**
system after it has been inserted? 1.2 The logical **structure** The Globe location service partitions a
www.cs.vu.nl/pub/papers/globe/asci.97.2.ps.Z

Locating Objects in Wide-Area Systems - van Steen, Hauck, Homburg.. (1998) (Correct) (14 citations)
Locating **Objects** in Wide-Area Systems Maarten van Steen
www4.informatik.uni-erlangen.de/~fzhauck/Pub/Doc/1998-01-commag.ps.gz

The Object System Pattern - Noble (1998) (Correct)
The **Object** System Pattern James Noble Microsoft Research
st-www.cs.uiuc.edu/~plop/plopd4-submissions/P55.ps

Exploiting Location Awareness for Scalable.. - Ballintijn, van.. (1999) (Correct) (1 citation)
Awareness for Scalable Location-Independent **Object** IDs Gerco Ballintijn Maarten van Steen Andrew S.
www.cs.vu.nl/pub/papers/globe/asci.99.ps.Z

A Scalable Implementation for Human-Friendly URIs - Ballintijn, Verkaik, Amade, .. (1999) (Correct)
it into two separate mappings and introduce the **object handle**. The first mapping is the HFN-to-**object**
two separate mappings and introduce the **object handle**. The first mapping is the HFN-to-**object handle**
resolution problem. The rest of this paper is **structured** as follows. Section 2 describes the
www.cs.vu.nl/pub/papers/globe/IR-466.99.ps.Z

An Approach to Behavior Sharing in Federated Database Systems - Doug Fang (1993) (Correct) (1 citation)
is described. In the context of a functional **object**-based database model, a technique to support
is not a limitation multiple arguments can be **handle**d by an obvious extension of our approach. 2 We
operations, methods, or functions) Liskov, 1988 **Strom** and Yemini, 1985]The primary concern of this
db.stanford.edu/pub/papers/iwdom.ps

Pickling State in the Java System - Riggs, Waldo, al. (1996) (Correct) (27 citations)
the Proceedings of the USENIX 1996 Conference on **Object**-Oriented Technologies Toronto, Ontario, Canada,
www.tns.lcs.mit.edu/~djw/library/coots96-riggs.ps.gz

Behaviour Specification in Database Interoperation - Vermeer, Apers (1997) (Correct) (2 citations)
locally implemented behaviour in a federation of **object**-oriented databases. In particular, given a
wwwis.cs.utwente.nl:8080/isdoc/confpaper/vermeer.caise97.accepted.ps.gz

Algorithmic Design of the Globe Location Service: Basic .. - Hauck, van Steen.. (1996) (Correct)
3, 1996 Abstract In Globe, a distributed shared **object** provides one or more contact addresses to
a globally unique, and location-independent **object handle**. The location service is capable of returning
our algorithms. In Section 5, we describe the data **structures** and some global consistency rules. The
www4.informatik.uni-erlangen.de/~fzhauck/Pub/Doc/1996-12-IR-413.ps.gz

Solitonic Strings and BPS Saturated Dyonic Black Holes - Cvetic, Tseytlin (1995) (Correct) (2 citations)
/ In N ,it is important that the oscillating **object** should be a **string**-like, i.e. having an
string sources should probably involve a `thin **handle**'type resummation of **string** loop expansion [13]
hep-th/9512031 December 1995 Solitonic **Strings** and BPS Saturated Dyonic Black Holes Mirjam
preprints.cern.ch/archive/electronic/hep-th/9512/9512031.ps.gz

Algorithmic Design of the Globe Wide-Area Location Service - van Steen, Hauck (1997) (Correct) (10 citations)

of a worldwide location service for distributed **objects**. A distributed **object** can reside at multiple
www.cs.vu.nl/pub/papers/globe/IR-440.97.ext.ps.Z

KITP-93: An Automated Inference System for Program Analysis - Wang, Goldberg (1994) (Correct)
11. y: ff)x =y=x: ff) manual-rule] 3 Proof **Objects** KITP-93 provides inference service through a
G: Thus, the condition of a rewriting rule can be **handle**d similarly as the subgoals (literals) inherited
the following statement into the KB, 1. 8(s)**string**p(s) 8(k: char)k in s)k 2 k 7)
ftp.kestrel.edu/pub/papers/goldberg/goldberg-cade-94.ps

Coordinating Distributed Objects With Declarative Interfaces - Narinder Singh (1995) (Correct) (9 citations)
Coordinating Distributed **Objects** With Declarative Interfaces Narinder P. Singh
cuiwww.unige.ch/OSG/people/jvitek/Resources/Archive/oopslaSingh.ps.gz

Towards Object-based Wide Area Distributed Systems - v.Steen, Homburg, van.. (1995) (Correct) (7 citations)
Towards **Object**-based Wide Area Distributed Systems Maarten van
www.cs.vu.nl/~philip/papers/iwooos95.ps.Z

A Model for Worldwide Tracking of Distributed Objects - van Steen, Hauck, Tanenbaum (1996) (Correct) (5 citations)
A Model for Worldwide Tracking of Distributed **Objects** Maarten van Steen, Franz J. Hauck, Andrew S.
www.cs.vu.nl/pub/papers/globe/tina.96.ps.Z

Semistructured Data - Buneman (1997) (Correct) (123 citations)
biologists [36]Superficially it looks like an **object**-oriented database system, for it has a schema
disparate databases. Third, even when dealing with **structured** data, it may be helpful to view it as
technology. Some of these, such as documents with **structured** text [3, 2] and data formats [9, 17]while
ftp.cis.upenn.edu/pub/papers/db-research/semistructured-paper.ps.Z

Discontinuous Dependency Parsing of Free and Fixed Word Order.. - Covington (1994) (Correct) (1 citation)
Give an example the students. The indirect **object** of the verb always comes before the direct
of Covington (1987, 1990, 1992) can be extended to **handle** partly or completely fixed word order, while
to parse left-branching and right-branching **structures** in less stack space than center-embedded
www.coling.uni-freiburg.de/~neuhaus/papers/covington/ai199402.ps.gz

Object Interconnections - Distributed Callbacks (Correct)
Object Interconnections Distributed Callbacks and
callback is the function pointer passed to set new **handle**r. If a new **handle**r callback has been installed,
www.iona.com/hyplan/vinoski/col8.ps.Z

First 20 documents Next 20

Try your query at: Amazon Barnes & Noble Google (RI) Google (Web) CSB DBLP

Search Results

Search Results for: **[(heap string)]**
Found **3** of **114,152 searched.**

Search within Results

> Advanced Search

> Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 3 of 3 short listing

1 An editor for revision control 80%
Christipher W. Fraser , Eugene W. Myers
ACM Transactions on Programming Languages and Systems (TOPLAS) March 1987 Volume 9 Issue 2
Programming environments support revision control in several guises. Explicitly, revision control software manages the trees of revisions that grow as software is modified. Implicitly, editors retain past versions by automatically saving backup copies and by allowing users to undo commands. This paper describes an editor that offers a uniform solution to these problems by never destroying the old version of the file being edited. It represents files using a generalization of AVL trees calle ...

2 REDUCE/1700: A micro-coded Algebra system 77%
Martin L. Griss , Robert R. Kessler
Proceedings of the 11th annual workshop on Microprogramming November 1978
The status of an ongoing micro-coded Algebra machine project is reviewed. We have implemented a LISP “machine” on the Burroughs B1726 computer, capable of supporting the REDUCE Algebra system. A portable version of this LISP machine (written in a portable implementation language, BIL), can be used to produce a compact and efficient LISP or REDUCE for smaller machines (it also serves as a bootstrapping kernel for larger machines). In this paper, we summarize the curren ...

3 UPC performance and potential: a NPB experimental study 77%
Tarek El-Ghazawi , Francois Cantonnet
Proceedings of the 2002 ACM/IEEE conference on Supercomputing November 2002
UPC, or Unified Parallel C, is a parallel extension of ANSI C. UPC follows a distributed shared memory programming model aimed at leveraging the ease of programming of the shared memory paradigm, while enabling the exploitation of data locality. UPC incorporates constructs that allow placing data near the threads that manipulate them to minimize remote accesses This paper gives an overview of the concepts and features of UPC and establishes

accesses. This paper gives an overview of the concepts and features of UPC and establishes, through extensive performance measurements of NPB work ...

Results 1 - 3 of 3 short listing

CiteSeer Find: memory management and string and Documents Citations

Searching for **memory management and string and heap**.
Restrict to: Header Title Order by: Citations Hubs Usage Date Try: Amazon B&N Google (RI) Google (Web) CSB DBLP
35 documents found. **Order: citations weighted by year.**

The Swift Java Compiler: Design and Implementation - Scales, Randall, Ghemawat, Dean (2000) (Correct) (3 citations)
other languages. In addition, its automatic **memory management** takes care of a time-consuming aspect of technology that is relevant to the technical **strategy** of the Corporation and has the potential to efficient code. The required run-time checks and **heap** allocation of all objects can introduce
ftp.digital.com/pub/Digital/WRL/research-reports/WRL-TR-2000.2.ps.gz

On gaining efficiency in completion-based theorem proving - Hillenbrand, Buch, Fettig (1996) (Correct) (7 citations)
Using these atterms with free{list based **memory management** allows us to dispose terms in constant into an inference machine, and sophisticated control **strategies**, all that combined with space saving of memory. We realized this **strategy** with a regular **heap**. The wish to remove an orphan immediately after
www.mpi-sb.mpg.de/~hillen/documents/HBF96.ps

Persistent Operating System Support for Java - Dearle, Hulse, Farkas (1996) (Correct) (5 citations)
are executing. This considerably simplifies **memory management** and adds a degree of inter-thread tables [1] package. However, the mappings may have **string**, integer, boolean and capability attributes threads, 4. shared libraries, and, 5. at least one **heap** containing Java objects. Grasshopper enables this
research.sun.com/research/forest/UK.Ac.Gla.Dcs.PJW1.Al_Dearle2_ps.ps

Automatic Removal of Array Memory Leaks in Java - Shaham, Kolodner, Sagiv (2000) (Correct) (1 citation)
have the following drawbacks: Explicit **memory management** complicates program logic and may lead to A standard Java implementation of a stack data **structure** is shown in Figure 1(a)After a successful in many Java applications. Our measurements of **heap** size show improvement on some example programs. 1
www.math.tau.ac.il/~rans/cc00.ps.gz

Efficient Object Sampling Via Weak References - Agesen, Garthwaite (2000) (Correct) (1 citation)
ABSTRACT The performance of automatic **memory management** may be improved if the policies used in collectors. 1.1 Improving Generational Collectors **Str**ongly typed languages like the Java TM collection services typically allocate objects in a **heap**. Periodically, the collector locates the set of
www.cs.purdue.edu/homes/hosking/ismm2000/papers/garthwaite.pdf

WALDMEISTER: High Performance Equational Deduction - Hillenbrand, Buch, Vogt.. (1997) (Correct) (3 citations)
only. In conjunction with free-list based **memory management**, we can dispose of terms in constant time. operations on the lowest level, where we put great **stress** on efficient data **structures** and algorithms. For data **structure** for storing critical pairs, a **heap** of **heaps**. Thereby, between 60 and 90 %of all
www.mpi-sb.mpg.de/~hillen/documents/HBVL97.ps

Unlimp - Uniqueness as a Leitmotiv for Implementation - Kahrs (1992) (Correct) (7 citations)
waste of space, but it also has advantages: **memory management** becomes easy, and sharing analysis [22] :EG !V G and t G :EG !V G ,assigning a **string** of source vertices and a **string** of target by using hash-consing for the creation of **heap** objects. We investigate the consequences of
www.cs.ukc.ac.uk/pubs/1992/575/content.ps.gz

The Bits Between The Lambdas: Binary Data in a Lazy.. - Wallace, Runciman (1998) (Correct) (2 citations)
same API -a new and useful abstraction over **memory management** and file management. This uniformity of for treating storage media as arbitrary-length **streams** of bits, without byte-alignment constraints. So data **structures**, whose operations provide both in-**heap** data compression and convenient high-level binary
ftp.cs.york.ac.uk/pub/malcolm/ismm98.ps.gz

An experimental study of compression methods for.. - Iivonen, Nilsson, Tikkanen (1999) (Correct) (1 citation)
heap supporting automatic disk-backed **memory management** in a soft real-time environment. Shades an ideal choice for a functional main-memory index **structure**. Keywords functional data **structures**,

in C on top of Shades [23]a persistent functional **heap** supporting automatic disk-backed **memory management**
hibase.cs.hut.fi/waaapl99.ps

Operating System support for Java - Dearle, Hulse, Farkas (1996) (Correct) (3 citations)
are executing. This considerably simplifies **memory management** and adds a degree of inter-thread
However, the mappings may be associated with **string**, integer, boolean and capability attributes.
3. stacks for threads, 4. at least one **heap** containing Java objects. Grasshopper enables
persistence.cs.stir.ac.uk/pub/papers/OS-support-Java.ps.gz

Formal Models of Distributed Memory Management - Ungureanu, Goldberg (1996) (Correct) (2 citations)
Formal Models of Distributed **Memory Management** Cristian Ungureanu and Benjamin Goldberg
programs. Programs have both the "code" control **string**) and the "store" syntactically apparent.
simple local garbage collector which scans a local **heap** starting from the local "stack" and the "incoming
www.cs.nyu.edu/phd_students/ungurean/tr728.ps

Monet And Its Geographic Extensions: a Novel Approach to.. - Boncz, Quak, Kersten (1996) (Correct) (1 citation)
large data. Monet provides many options in **memory management** and virtual-memory clustering **str**ategies to
in **memory management** and virtual-memory clustering **str**ategies to optimize access to its tables. We
by Monet's flexible **memory management** using **heaps**. The extra cost for re-assembling multiattribute
www.wins.uva.nl/research/isis/pub/sequoia.ps.gz

Software—Practice And Experience, Vol. 24(6), 565–578 (june .. - Design Of Safe (Correct)
corrupt on-**heap str**uctures used by dynamic **memory management**, manifesting later (often in an unrelated
VOL. 24(6)565-578 (JUNE 1994) Design of a Safe **String** Library for C ajith k. narayanan AVL List GmbH,
www.cs.ubc.ca/local/reading/proceedings/spe91-95/spe/./vol24/issue6/spe898.pdf

Design and Implementation of a Distributed Crawler And .. - Zeinalipour-Yazti.. (2002) (Correct)
caching of crawling state, customized **memory management**, employment of persistent data **str**uctures
memory management, employment of persistent data **str**uctures with disk-caching support, optimizations of
computing nodes, execute in different Java **heap** spaces, and communicate through a permanent socket
www.cs.ucr.edu/~csyiazti/downloads/papers/ngits02/ngits02.pdf

Interprocedural Compatibility Analysis for Static.. - Gheorghioiu.. (2003) (Correct)
Processors|compiler, optimization, **memory management** (garbage collection) General Terms
many unitary allocation sites allocate exception, **string** buer, or iterator objects. We identify two
all objects are allocated in a garbage-collected **heap**. While this abstraction simpli es many aspects of
www.cag.lcs.mit.edu/~rinard/paper/popl03.ps

Establishing Local Temporal Heap Safety Properties.. - Shaham, Yahav.. (Correct)
Properties with Applications to Compile-Time **Memory Management** Ran Shaham 1,2 Eran Yahav 1 Elliot
of a singly-linked list public static void main(**String** args[L x, y, t 1] x =null 2] while
Establishing Local Temporal **Heap** Safety Properties with Applications to
www.cs.tau.ac.il/~yahave/papers/sas03-safety-mm.ps

WALDMEISTER: High performance equational theorem proving - Buch, Hillenbrand, Fettig (Correct)
atterms in conjunction with free{list based **memory management** allows us to dispose terms in constant
machine, and at the top the overall control **str**ategy guiding the search for promising derivations.
Topped by a two level data **str**ucture, basically a **heap** of **heaps** allowing to delete between 60 %and 90 %
www.mpi-sb.mpg.de/~hillen/documents/BHF96.ps

Write Barrier Removal by Static Analysis - Zee, Rinard (Correct)
garbage collectors have become the **memory management** alternative of choice for many safe
that updates an intergenerational reference data **str**ucture. This data **str**ucture enables the garbage
references |at every instruction that stores a **heap** reference into an object, the compiler inserts
www.lcs.mit.edu/publications/pubs/ps/MIT-LCS-TR-834.ps

First 20 documents Next 20

Try your query at: Amazon Barnes & Noble Google (RI) Google (Web) CSB DBLP

CiteSeer Find: memory management and string and Documents Citations

Searching for **memory management and string and heap**.
Restrict to: Header Title Order by: Citations Hubs Usage Date Try: Amazon B&N Google (RI) Google (Web) CSB DBLP
35 documents found. **Order: citations weighted by year.**

Region-based Memory Management for Real-time Java - Higuera, Issarny, Banatre.. (2001) (Correct)
Region-based **Memory Management** for Real-time Java Teresa Higuera,
class RegionUseExample {public static void main (**String**[args) ScopedMemory myRegion =new
3 java.sun.com/products/cldc/wp/ 1 the **heap**, and another of 20 integers in the memory region
www-rocq.inria.fr/arles/doc/ps01/isorc01.pdf

Real-Time Garbage Collection in a - Multimedia Programming Language (Correct)
Position Paper OOPSLA 1993 workshop on **Memory Management** and Garbage Collection September 19, 1993
as well as regular **heap** allocated C data such as **str**uctures and arrays, we must deal with several
collection for both OIC objects as well as regular **heap** allocated C data such as **str**uctures and arrays, we
ftp.cs.utexas.edu/pub/garbage/GC93/hennessey.ps

A Method for Automatic Optimization of Dynamic.. - Häggander.. (Correct)
A Method for Automatic Optimization of Dynamic **Memory Management** in CDaniel Hggander, Per Lidn and Lars
the runtime behavior, where the same object **str**uctures tend to be created and used over and over
www.ide.hk-r.se/~dha/icpp-01.ps

Escape Analysis for Stack Allocation in Java - Eun-Sun Cho And (Correct)
Garbage collecting objects in Java makes **memory management** easier for the programmer, but it is time
behavior is based on the concept of procedure **string** in Harrison's work[4]He proposed an escaping
in loops or in recursive functions. 2 ORef in **heap**, environment and escaping is the set of ideals
pllab.kaist.ac.kr/~kwang/paper/00-ecoop-chyi.ps.gz

High-Performance Crawling and Filtering in Java - Zeinalipour-Yazti, Dikaiakos (2001) (Correct)
support for multithreading, customized **memory management**, employment of persistent data **str**uctures
memory management, employment of persistent data **str**uctures with disk-caching support, optimizations
dierent computing nodes, execute in dierent Java **heap** spaces, and communicate through a permanent socket
www.cs.ucy.ac.cy/mdd/publ/TR-2001-3.ps.gz

HCL - a Language for Internet Data Acquisition - Richard Connor And (Correct)
some simple experiments in which simple **memory management** is built into the **string** implementation,
the problem domain. Our primary aim is to allow the **str**aightforward automation of tasks currently
functions, implemented in a garbage-collected **heap**. Characters are represented as **strings** of length
www.cs.strath.ac.uk/~hippo/papers/hcl.ps

A Conservative Garbage Collector for an EuLisp to ASM/C.. - Ulrich Kriegel Fraunhofer (Correct)
a 4 byte tag in front of data otherwise. The **memory management** system for the EuLisp runtime system relies
et al.1992]at ISST we are investigating **str**ategies for the compilation of EuLisp modules [
collection more than one third of the allocated **heap** is still in use then the **heap** size will be
ftp.cs.utexas.edu/pub/garbage/GC93/kriegel.ps

Representing Polynomials in Computer Algebra Systems - Apel, Klaus (Correct)
on the data, and the description of the **memory management** for this data type. The result of such an
designed for computations in and with algebraic **str**uctures and substructures. The basic domains
Figure 1: FELIX memory map 6 available memory **heap** node cells rational number cells long integer
www.informatik.uni-leipzig.de/~apel/publications/pereslavl.ps

On the Type Accuracy of Garbage Collection - Hirzel, Diwan (2000) (Correct)
and compare them to the original explicit **memory management** in the C benchmark programs. We use a
what that address maps to when interpreted as a **string**, int, long, or float. In other words, this table
in all regions of memory (globals, locals, and **heap**)A conservative garbage collector cannot reliably
www.cs.colorado.edu/~diwan/ISMM-Hirzel.ps

Prop - Language Reference Manual - Leung (1997) (Correct)
: 27 4.5 **Memory management** :
Institute of Mathematical Sciences 251 Mercer **Stree**t New York, NY 10012 April 4, 1997 Abstract This
www.cs.nyu.edu/leunga/www/refman.ps

Adding Persistence to the Oberon-System - Knasmüller (1996) (Correct)
Objects 3.3 Loading Objects 3.4 Persistent **Memory Management** 3.5 Programming Interfaces 3.6 Necessary
The following code shows how to make a **string** object (identified by the key myroot)
in the Oberon system is obtained by a persistent **heap** on the disk. Persistent objects are on this **heap**,
ftp.ssw.uni-linz.ac.at/pub/Reports/Report6.ps.Z

Global Regions - Holds Values (Correct)
Tofte and Jean-Pierre Talpin. Region-based **memory management**. Information and Computation,
183 StatObject, 206 storage mode, 100 **str**, 48 **String**.h, 190 **str**ongly connected component,
www.cs.cmu.edu/afs/cs/user/birkedal/pub/manual.ps.gz

A Win32 Programming Interface for SML/NJ - Liang, Huelsbergen (1995) (Correct)
services (processes and threads, file I/O, **memory management**, etc. 1 security, multimedia, and
C, programmers are able to take advantage of ML's **str**ong static typing, higher-order functions, the
Win32 functions frequently take pointers into the C **heap** as arguments, and return C **strings** and **str**uctures.
www.cs.yale.edu/users/liang-sheng/smlwin32.ps.gz

Reference Manual (Version 1.1) - Guy Blelloch (Correct)
front end that handles program control and **memory management**, and specialized back ends that implement
to VCODE to gain full performance. **Memory management Memory management** is a major problem in designing
parallelism being useful for only a small class of **str**uctured applications led to its virtual exclusion
www.cs.cmu.edu/~scandal/papers/CMU-CS-91-146.ps.gz

Documents 21 to 35 Previous 20

Try your query at: Amazon Barnes & Noble Google (RI) Google (Web) CSB DBLP